

B

The Mousetrap Programmer's Guide

1. The Mousetrap Effect.....	1
2. Operation Overview.....	1
3. The Details (Overview).....	3
4. The Frameset Page.....	3
4.1. The .goTo and .startAt SML variables.....	3
4.2. The .acb SML variable.....	4
4.3. Constants.....	4
4.3.1. defaultStatusMsg.....	4
4.3.2. Hidden Frame Pages	4
4.3.3. defaultWarpTo	5
4.4. Variables	5
4.4.1. window.defaultStatus.....	5
4.4.2. acb	5
4.4.3. backToTop	5
4.4.4. linkClicked=false	6
4.4.5. warpTo=""	6
4.4.6. newWin=true.....	6
4.4.7. timerID=false	6
4.4.8. firstPageLoaded=false.....	6
4.4.9. enterClicked=false	6
4.4.10. initialLocation	7
4.5. The Functions.....	7
4.5.1. function unload().....	7
4.5.2. function clicked(aLink).....	8
4.5.3. function setStatus(msg).....	8
4.5.4. function loadPage()	9
4.5.5. Frameset Definition	9
4.5.6. NOFRAME	10
5. The Hidden Frame	10
5.1. The hidden1 File	10
5.1.1. if (parent.firstPageLoaded)	11
5.1.2. if (!parent.linkClicked)	11
5.1.3. if (!parent.backToTop).....	11
5.2. The hidden2 File	11
6. The Main Frame.....	12
6.1. The Warning Page.....	12
6.2. Sales Page	12
6.2.1. MAC.visit	12
6.3. Registration (Age Verification) Page.....	13
6.3.1. No loadPage()	13
6.3.2. isSent.....	13
6.3.3. Parent.newWin.....	13
6.3.4. The Submit Button.....	14

6.4. Survey Page.....	14
6.5. Allsites Registration Page	15
6.6. Softsales Page	15
6.7. Banners Page.....	16
6.7.1. Links	16
6.7.2. Unconditional Back/Warp	16
6.7.3. Conditional Back/Warp	17
7. Mousetrap Flow Drawing -- Flow Chart for Sex Roulette	18

The Mousetrap Programmer's Guide

1. The Mousetrap Effect

Normally the back button in a browser takes the user to the previous page. The "mousetrap" however, has the ability to redirect a user to a new URL when the back button is pushed. In addition if the user "warps" (change location via bookmarks, history list or typing in a new URL), a new browser window opens.

Lets look at a few senarios:

1. The user comes to the "Warning" page and hits the back button. They do not return to their previous URL but instead see a "soft" sales page (the photos are softcore). If they hit the back button again, they end up on a banners page. If they now decide to "warp" by going to their bookmarks, they will go the location they "warp" to but they will also open up a new browser which will overlay the warped-to browser window with yet another sales page.
2. The user comes to the "Warning" page and hits the enter button to go to a sales page. From the sales page, they click on a link and enter an age verification page. They do not have a credit card and decide to "warp" to a new location. Their browser opens up a new window with a "survey" page on it. They close the window with the survey and discover that the placed they warped to is now displaying in their original browser window.
3. The user comes to the "Warning" page and hits the enter button to go to a sales page. From the sales page, they hit the back button but instead of going back to the warning page, they see a "survey" page. If the user fills out the survey, they will get a free subscription to all sites. They fill out the survey and see the age verification page registration page. They fill out the form and enter the members area.

2. Operation Overview

The core function of the mousetrap is achieved through the use of a hidden frame and JavaScript. All pages in the mousetrap are displayed within a frameset which has two frames, a frame named "main" which fills up 100% of the frameset and a frame named "hidden" which is indeed hidden.

When the user clicks on a link (like the "Enter" on a warning page), the main frame (no pun intended) is loaded. While the browser is loading the main frame (with a sales page), it starts to execute the JavaScript on the sales page. This JavaScript runs the "loadPage()" function which then loads into the hidden frame, the "hidden1" page. The "hidden1" page then executes its JavaScript which then loads the "hidden2" page.

NOTE: To install a Mousetrap, you *must* read "The Mousetrap for HTML Designers".

This is the sequence of events for a link click:

Link Click Events (Click Enter on Warning Page)¹

Event#	Description	Frameset	JavaScript ²
1	Click Enter on Warning page	main	parent.clicked()
2	Load Sales1 Page	main	parent.loadPage()
3	Load Hidden1	hidden	loadPage() (in hidden1)
4	Load Hidden2	hidden	—

Notice that the last page loaded in the browser's history list is the "hidden2" page (hidden1 and hidden2 are frameset variables which hold the URL's of the documents loaded into the "hidden" frame).

When the user presses the browser's "back" button the hidden2 page is unloaded and the hidden1 page is loaded. The hidden1 page runs its "loadPage()" function which then loads a page into the "main" frame. The user will not return to the warning page but instead will see a second sales page in the "main" frame. The Sales2 page will then load the Hidden2 page into the hidden frame:

This is the sequence of events caused by pressing the browser's back button:

Back Button Event (Backing from First Sales Page)¹

Event#	Description	Frameset	JavaScript ²
1	Browser's back button clicked	—	—
2	Load Hidden1	hidden	loadPage() (in hidden1)
3	Load Sales2 Page	main	parent.loadPage()
4	Load Hidden2	hidden	—

From the second sales page, if the user decides to "warp" by going to Yahoo via a bookmark, the frameset page itself will unload. When the frameset unloads, its "onUnload" event causes the execution of its "unload()" function. This function creates a new window with yet another sales page in it (perhaps a page of banners).

This is the sequence of events caused by "warping" to Yahoo:

Jumping to new URL via Bookmark¹

Event#	Description	Frameset	JavaScript ²
1	Original window loads Yahoo	main	frameset's "unload()"
2	New Window loads hidden1 page and banners page	hidden	—
3	Load Hidden2	main hidden	parent.loadPage() —

Note 1: At the start of events, the "hidden2" page is currently in the "hidden" frame.

Note 2: This is the JavaScript function which initiates the next event.

3. The Details (Overview)

The next three sections explain the functions and interactions of the frameset, main, and hidden pages. See Appendix A for flowchart drawing of Sex Roulette sales site (as of February 9, 1998).

This section assumes a familiarity with SML (see *SML for Dummies* at the following URL: http://kona.sbusiness.com/sml_for_dummies/). This section also assumes familiarity with JavaScript.

Note: The mousetrap mechanism allows the normal use of sales pages if the browser has JavaScript turned off or is not JavaScript enabled. It is even possible to access the sales pages if the browser is not "frames" enabled. In these cases, the back button will operate normally.

Netscape 2 executes the back button at the frameset level and removes the entire frameset. This means that the "back" button acts like a "warp" in Netscape 2.

4. The Frameset Page

The frameset page contains the JavaScript functions and variables accessed by the various pages loaded into the "main" and "hidden" frames. The beauty of frameset level variables, is that state can be maintained between pages. Typically this page is named something like "mmuse_index.sml" or "mass_index.sml" (the first "m" stands for mousetrap).

We will go through the frameset page line by line and provide comments.

4.1. The .goTo and .startAt SML variables

Normally the warning page is first loaded when the mousetrap starts up. However, the following URL would allow the starting at another page:

http://www.sexmuseum.com/?startAt=mmuse_sales1.sml

The first of the following lines allows that to happen (they must be in this order):

```
<$if cond:.startAt,cond:!.goTo val:.goTo=.startAt><$endif>  
<$if cond:!.goTo val:.goTo="mmuse_warning.sml"><$endif>
```

It might be easiest to examine the overall effect of these two lines by looking at the "goTo" defined condition:

Condition1 - goTo not defined: If "startAt" is defined, "goTo" is assigned to it. Otherwise the warning page is loaded.

Condition2 - goTo defined: When the user warps, a new window is opened with a URL that may look like the following:

http://www.sexmuseum.com/?bid=acb100002-60000&goTo=mmuse_survey.sml

The "goTo" variable is used to in the frameset definition to load a page into the main frame.

4.2. The .acb SML variable

Typically the sales sites are entered through the clicking of a banner. That banner includes in its URL a parameter to identify the person and banner. The following URL will load the sex museum sales page:

```
http://www.sexmuseum.com/?acb=acb100001-g1400
```

In the "acb" parameter (ad campaign banner), the 100001 is the advertiser number and the g1400 is the number corresponding to a particular banner.

Raw hits are assigned to an advertiser whenever the server notices a page loaded with the "acb" parameter in the URL. With the following line:

```
<$if cond:.acb val:.bid=.acb><$endif>
```

If the .acb variable exists (not everyone comes from a banner), the SML variable .bid is assigned to its value. On subsequent pages, the advertiser number is passed by appending the following to the URL:

```
?bid=acb100001-g1400
```

We don't want to append "?acb=acb100001-g1400" because this could inadvertently cause another raw hit to be assigned to the advertiser.

4.3. Constants

4.3.1. defaultStatusMsg

Each page sets the default message by loading the following variable:

```
defaultStatusMsg="Welcome to Sex Museum";
```

4.3.2. Hidden Frame Pages

The hidden frame gets loaded with the URLs of the following variables.

```
hidden1="mmuse_hidden1.sml"; //Change in frameset definition too  
hidden2="mmuse_hidden2.sml";
```

Note that the URL at hidden1 must also be set in the frameset definition.

4.3.3. defaultWarpTo

It appears that Netscape 2 cannot retain variables in its parent frame when going from page to page.

```
defaultWarpTo="mmuse_survey.sml";//Kludge for Netscape 2 - warp  
location in registration off sales 1
```

The registration (age verification) pages can be reached from different sales pages (the softsales and regular sales pages may use the same registration pages). Because of this, the registration pages do not re-define the "warpTo" or "backTo" variables and do not run the parent.loadpage() function as well. This allows a normal "back" browser function to occur and the user can return to the previous sales page.

With Netscape 2, if someone warps out of a registration page, the parent.warpTo variable contains garbage. If there is garbage in the "warpTo" variable, the "defaultWarpTo" value is used instead. This variable is used in the "unload()" function (see section 0).

4.4. Variables

Each page loaded into the "main" frame sets the window's "defaultStatus with the following variable:

4.4.1. window.defaultStatus

This sets the default status message for the current window, which is the frameset page.

4.4.2. acb

The following loads the "acb" variable with the current banner ID:

```
acb ="<#.bid#>"
```

This variable is checked in the softsales page to decide whether to send the user to our banners page. If the user has come from Yahoo, for example, they won't have a banner ID and they may also be under age so they are not sent our banners page.

4.4.3. backToTop

When the user hits the back button, they normally load a new page within the current frameset. The following line sets this default:

```
backToTop=false;
```

In the "banners" page, however, this variable is set to true (parent.backToTop=true). This allows the user to back out of the banners page and totally exit the mousetrap: the frameset page gets unloaded and replaced with a new page (the mousetrap's last gasp).

4.4.4. linkClicked=false

When a linked is clicked this variable is set to true.

The sequence of loading the main frame and the hidden frame pages varies, depending on the setting of this variable. This variable is set in the "clicked(aLink)" function and used in a function defined in this section ("loadPage()") and in the hidden1 page.

4.4.5. warpTo=""

This variable is set by the various pages loaded into the main frame. If the user tries to leave the sales pages via bookmarks, history list, or just typing in a URL, the user will end at the URL/file pointed to by this variable.

4.4.6. newWin=true

The setting of this variable determines if we should create a new window at warp time. This is normally set to true except when the submit button is clicked on the registration page. After the submit button is pressed, the user may get presented with a page that has a link to the members area. This link has the "target=_top" attribute and we don't want to create a new sales window when the frameset is blown away as the user tries to enter the members area.

4.4.7. timerID=false

This variable is used in the "setStatus(msg)" function to indicate if a timer is running.

4.4.8. firstPageLoaded=false

This variable is used and set in the hidden1 page. Normally hidden1 would load hidden2 but the first time hidden1 is loaded, hidden1 does nothing because this variable is set to false. This allows the first page loaded into the main frame to load hidden2 and thus guaranteeing that hidden2 will load *after* the first main frame page.

4.4.9. enterClicked=false

This variable is currently used on the Sex Museum site. It is set to true on the "mmuse_survey.sml" page. If the user gets to the survey page and then hits the back button, they will end up on the banners page. It is also possible to get to the banners page by hitting the back button from the warning page. For users who have gotten to the banners page and have clicked "Enter" on the warning page, we drop them into the sales pages of another site.

4.4.10. initialLocation

The following initializes the initialLocation variable:

```
i=location.href.indexOf("?acb=");  
if (i==-1) initialLocation = location.href;  
else initialLocation = location.href.substring(0,i) + "?bid" +  
    location.href.substring(i+4,location.href.length);
```

This variable is used in the "unload()" (see section 0). Basically the above lines change the "acb" parameter name in a URL to "bid" if the "acb" exists.

For example:

http://www.sexmuseum.com/mmuse_index.sml?acb=acb100002-60000

becomes:

http://www.sexmuseum.com/mmuse_index.sml?bid=acb100002-60000

If the user warps out and a new window with a new frameset is loaded, we need to change "acb" to "bid" because it might be possible to give another payed click to the same advertiser (the server looks for "acb=acb..." in a URL to assign a visit to an advertiser).

4.5. The Functions

4.5.1. function unload()

This function runs when the frameset page is unloaded caused by the user attempting to warp away from our sales site.

```
if ((warpTo.indexOf(".sml") == -1) && (warpTo.indexOf(".htm") == -1)  
    && (warpTo.indexOf("http") == -1)) warpTo = defaultWarpTo;
```

The above lines are used to get around a limitation in Netscape 2 JavaScript. Variables set in the frameset (parent) page are not retained when a new page is loaded into the main frame. See "defaultWarpTo" in section 4.3.3 for more information. A corrupted "warpTo" is detected by checking to see if it has the ".sml", ".htm", or "http" strings in it.

```
if ((window.name != "newWindow") && newWin)
```

A new window will only be created at warp time if we are not already in a new window and the "newWin" is true. We only create a new window at warp time only once so the user can leave the sales site the second time they warp out.

```
{  
    if (backToTop)  
        window.open(warpTo, 'newWindow');
```

If "backToTop" is true, the frameset page will be replaced with the URL indicated by the "warpTo" variable and we will have exited the mousetrap.

```

else
{
    if(initialLocation.indexOf("?") == -1)
        initialLocation=initialLocation + "?";
    else initialLocation=initialLocation + "&";
    window.open(initialLocation + 'goTo=' + warpTo, 'newWindow');
}

```

Next the function checks to see if “initialLocation” has a “?”, if not it adds one, otherwise it adds a “&”. Finally, we reload the frameset page with the “goTo” parameter appended to the end of the initial frameset’s URL.

4.5.2. function clicked(aLink)

This function needs to run every time a link is clicked. Every link should invoke this method via the “onClick” event.

```

linkClicked=true;
if (aLink.target && (aLink.target.indexOf('_top') == 0))
    newWin = false;

```

This function sets “linkClicked” and then checks to see if the link’s target attribute is set to “_top”. If so, we don’t want to create a new window when this link is clicked (“newWin” is used in the “unload()” function [see section 0]).

4.5.3. function setStatus(msg)

This function is called from within all pages that wish to set the status message when the user places the mouse pointer over a link.

```

top.main.status=msg;
if (timerID) clearTimeout(timerID);
if (msg != '') timerID=setTimeout("window.status='',5000");
return true;

```

The timer function is used because Internet Explorer does not support a onMouseOut event. The timer will automatically clear the status message after 5 seconds. If the timer is running when this function is entered, it gets initially cleared before being set again.

4.5.4. function loadPage()

This function along with the function in hidden1 contain the logic to control the back button's action.

```
if (!firstPageLoaded) {setTimeout('loadPage()',500);return}
```

The warning page is the first page to call "loadPage()." If hidden1 has not been loaded (indicated by "firstPageLoaded" being false), the warning page cannot load hidden2. We want to be sure that hidden1 has been loaded before hidden2 is loaded (otherwise there will be no browser history of hidden1 preceding hidden2).

The above code starts a cycle of calling the "loadPage()" (i.e. calling itself) every 1/2 second until hidden1 is loaded. Once hidden1 is loaded (and it sets "firstPageLoaded to false), "loadPage()" can continue to the next section:

```
if(!parent.linkClicked)
    parent.hidden.location.href=hidden2 + "<#SID=.#>";
else
    parent.hidden.location.href=hidden1 + "<#SID=.#>&bid=<#.bid#>";
```

In the first case, "linkedClicked" is false. This means that the current page in the main frame (the one calling this function) was not loaded through a link click. That is it was loaded by the "loadPage()" function in hidden1 (which means the browser's back button was pressed). Therefore the hidden2 must be loaded in the hidden frame. See "Operation in a Nutshell" page 1 for details in the sequencing of a browser's back button press.

In the second case, the current page in the main frame (the one calling this function) was loaded through a link click. This means hidden1 must be loaded in the hidden frame. Hidden1 will then load hidden2. See "Operation in a Nutshell" page 1 for details in the sequencing of a link click.

4.5.5. Frameset Definition

The frameset definition sets up a page with two frames, one taking up 100% of the frameset and the other taking up the remainder (which is nothing so the frame is hidden).

```
<FRAMESET ROWS="100%,*" SCROLLING=no BORDER=0 frameborder=no
framespacing=0 onUnload="unload()">
```

```
<FRAME SRC="<#.goTo#><#SID=.#>&bid=<#.bid#>" NAME="main"
SCROLLING=auto BORDER=0 frameborder="0">
```

```
<FRAME SRC="mmuse_hidden1.sml?<#SID=.#>&bid=<#.bid#>"
NAME="hidden" SCROLLING=no BORDER=0 frameborder="0">
```

The main frame gets its source set from the .goTo SML variable. When the frameset is first loaded, this variable is set to the warning page. If there is a warp, this variable is set to whatever the "warpTo" variable was set to at warp time. The SML state ID and the banner ID parameters are also passed in this URL.

The hidden frame initially loads the hidden1 file, passing to it the SML state ID and the banner ID parameters.

4.5.6. NOFRAME

This section allows a non-frames browser access to the sales pages. The .goTo will take the user to the warning page or the warped to page.

```
<NOFRAME><HTML><HEAD>
<TITLE>Sex Roulette</TITLE></HEAD>
<BODY TEXT="#FFB000" BGCOLOR="#000000" LINK="#DD0000"
      VLINK="#DD0000" ALINK="#FFFF80">
Click <a href=<#.goTo#>>here</a> to continue...
</BODY>
</HTML>
</NOFRAME>
```

5. The Hidden Frame

The hidden frame is the secret behind the operation of the mousetrap. The hidden frame always loads last. Consequently when the browser's back button is pressed, the hidden2 file first unloads and then the hidden1 file gets loaded. Hidden1 has a JavaScript which then loads a page into the main frame.

5.1. The hidden1 File

This file is loaded into the hidden frame and is used to load the main frame when the back button is pressed.

```
function loadPage()
{
  if (parent.firstPageLoaded)
  {
    if (!parent.linkClicked)
    {
      if (!parent.backToTop)
        parent.main.location=parent.backTo +
          "?<#SID=.#>&bid=<#.bid#>";
      else
      {
        parent.newWin=false;
        top.location=parent.backTo;
      }
    }
    else
      parent.hidden.location=parent.hidden2;
    parent.linkClicked=false;
  }
  else parent.firstPageLoaded = true;
}
```

5.1.1. if (parent.firstPageLoaded)

We first check to see if this is the first time this file is being loaded. If so we do nothing except set the "parent.firstPageLoaded" variable to true. The first time this file is being loaded is when the frameset is being first loaded... the frameset loads this page into the hidden frame. Normally hidden1 would load the main frame but initially, this is being done by the frameset page.

5.1.2. if (!parent.linkClicked)

We need to check if this page got loaded by the pressing of the back button or by the "parent.loadPage()" being executed by the page in the main frame.

If the "parent.linkClicked" variable is true, then this page was loaded by the main frame page after a link click and we need to load the hidden frame with the hidden2 file.

Otherwise this page got loaded by the hitting of the browser's back button and we need to do further testing.

5.1.3. if (!parent.backToTop)

If "backToTop" is false, we will load the main frame with the "backTo" page. Otherwise we will replace the entire frameset with the "backTo" page.

5.2. The hidden2 File

The logical purpose of this page is mostly that of a placeholder... something to load after the main frame and hidden1 is loaded. The JavaScript in the body of this page is used strictly for testing purposes:

```
<script language="JavaScript">
  document.write("parent.backTo=" + parent.backTo + " | | | | | ");
  document.write("parent.warpTo=" + parent.warpTo + " | | | | | ");
</script>
```

To use the above in testing you have to modify the frameset page and the frameset definition itself near the bottom of the page. On the frameset page you should see the following:

```
<FRAMESET ROWS="100%," SCROLLING=no BORDER=0 frameborder=no framespacing=0
onUnload="unload()">
```

Change the frameset definition to the following for testing with the hidden2 file:

```
<FRAMESET ROWS="*,40" SCROLLING=no BORDER=0 frameborder=no framespacing=0
onUnload="unload()">
```

6. The Main Frame

The main frame contains all the pages the user views. Most pages have a similar JavaScript between the `</title>` and `</head>`.

6.1. The Warning Page

The warning page has the following JavaScript:

```
<SCRIPT LANGUAGE="JavaScript">
<!-- Script start
    parent.backTo="mxsp_softsales.sml";
    parent.warpTo="mxsp_softsales.sml";
    window.defaultStatus=parent.defaultStatusMsg;
    parent.loadPage();
// Script end -->
</SCRIPT>
```

The first two lines set the `backTo` and `warpTo` variables in the frameset page. The third line sets the default message. The fourth line starts the `loadPage()` function (which starts the sequence of loading into the hidden frame).

6.2. Sales Page

6.2.1. MAC.visit

The sales page has an SML tag on top of it:

```
<#MAC.visit#>
```

This SML tag causes a visit to be registered for the advertiser who's banner ID is appended to the URL as below:

```
http://www.xxxsexphotos.com/mxsp_index.sml?acb=acb100002-60000
```

When the user clicks the enter page and loads the page with the "MAC.visit" tag, the advertiser with the number "100002" gets paid for a click.

This tag should not be on the softsales page. We don't pay an advertiser when they attempt to back off the warning page.

The JavaScript for this page is identical to the warning page except for one additional line:

```
parent.enterClicked=true;
```

When this variable is true, the user has clicked the enter button on the warning page. The banners page may optionally look at this variable to decide where to redirect the user.

6.3.4. The Submit Button

The submit button, near the bottom of the page uses an SML conditional to decide whether to use a JavaScript “button” or a regular “submit” button. The variable “SID.js” is set to true in the “hidden2” page. This page could *only* be loaded if the user had JavaScript capable browser with JavaScript enabled. Therefore the code below will only use a JavaScript button if the user is running JavaScript:

```
<$if cond:SID.js>
<CENTER><input type="button" value="Submit Information"
onClick="SubmitData( this.form ) "></CENTER>
<$else>
<CENTER><input type="submit" value="Submit Information"></CENTER>
<$endif>
```

The SML conditional `<$if cond:MAC.user_agent LIKE "Mozilla/[2-9]">` should only be used on non-mousetrap registration pages. We have found about 5% of users have turned off JavaScript on a JavaScript capable browser. The condition searching for “Mozilla[2-9]” will present those users with a dead button and they will not be able to register.

6.4. Survey Page

This page currently takes a survey of users but never stores the data. In development is a survey page and Perl script which will store the data to SQL Server (this project was complete in early Feb, 1998).

This page combines the JavaScript of both the warning page (section 6.1) and the registration page (section 6.3). The submit button can be a JavaScript button because the user must have JavaScript to get to this page.

The fields in this form are all passed as parameters to the URL of the “allsites” registration page. The hidden fields are used to pass the “referrer”, the “bid”, and the “referer”. Because this is a submission, we have found it is possible to end up on a different server for the “allsites” page (especially when a Perl script “redirect” is eventually used to store the survey data). For this reason, the SID is useless and we have to directly pass the referrer as a parameter (the SID is specific to a server).

6.7. Banners Page

This page contains banners which promote Xpics' other sites. It may also contain banners of non-Xpics' sites.

6.7.1. Links

The banner links in this page all need the 'target="_top"' and the 'OnClick="parent.clicked(this)'" statements in all links. A sample link is shown below:

```
<a href="http://www.xpics.com/ass/ass.html?acb=acb100001-1100"
    Target="_top" OnClick="parent.clicked(this)">
```

6.7.2. Unconditional Back/Warp

This page has a line of JavaScript no used on previous pages:

```
<SCRIPT LANGUAGE="JavaScript">
    parent.backTo=
        "http://198.168.54.139/thrill-cgi/ad/xpics/ricochet.cgi?xpicsr";
    parent.warpTo=
        "http://198.168.54.139/thrill-cgi/ad/xpics/ricochet.cgi?xpicsr";
    parent.backToTop=true; //blow away frame
    window.defaultStatus=parent.defaultStatusMsg;
    parent.loadPage();
</SCRIPT>
```

The line "parent.backToTop=true" allows us to replace the frameset page with the pages at the backTo/warpTo URL's. This means the user will be exiting the mousetrap and is free to warp to wherever they wish. Of course, if they hit the back button again, they will be back to the mousetrap.

6.7.3. Conditional Back/Warp

This page can optionally have a conditional in the JavaScript. If the person gets to this page via the Survey page, they have already clicked the enter button on the warning page. This means that we can drop them to a sales page on another Xpics site when they try to back/warp. The JavaScript for this page follows:

```
<SCRIPT LANGUAGE="JavaScript">
  if (!parent.enterClicked)
  {
    parent.backTo=
      "http://198.168.54.139/thrill-cgi/ad/xpics/ricochet.cgi?xpicsr";
    parent.warpTo=
      "http://198.168.54.139/thrill-cgi/ad/xpics/ricochet.cgi?xpicsr";
  }
  else
  {
    parent.backTo="http://www.videosexchannels.com/vsc_sales.sml?bid=
    <#.bid#>&refer=MMUSE_banners.sml-<#MAC.NBHOST#>";
    parent.warpTo="http://www.videosexchannels.com/vsc_sales.sml?bid=
    <#.bid#>&refer=MMUSE_banners.sml-<#MAC.NBHOST#>";
  }
  parent.backToTop=true; //blow away frame
  window.defaultStatus=parent.defaultStatusMsg;
  parent.loadPage();
</SCRIPT>
```

Notice in the JavaScript, the "refer" variable is being passed to the "vsc_sales.sml" page. This accomplishes two things:

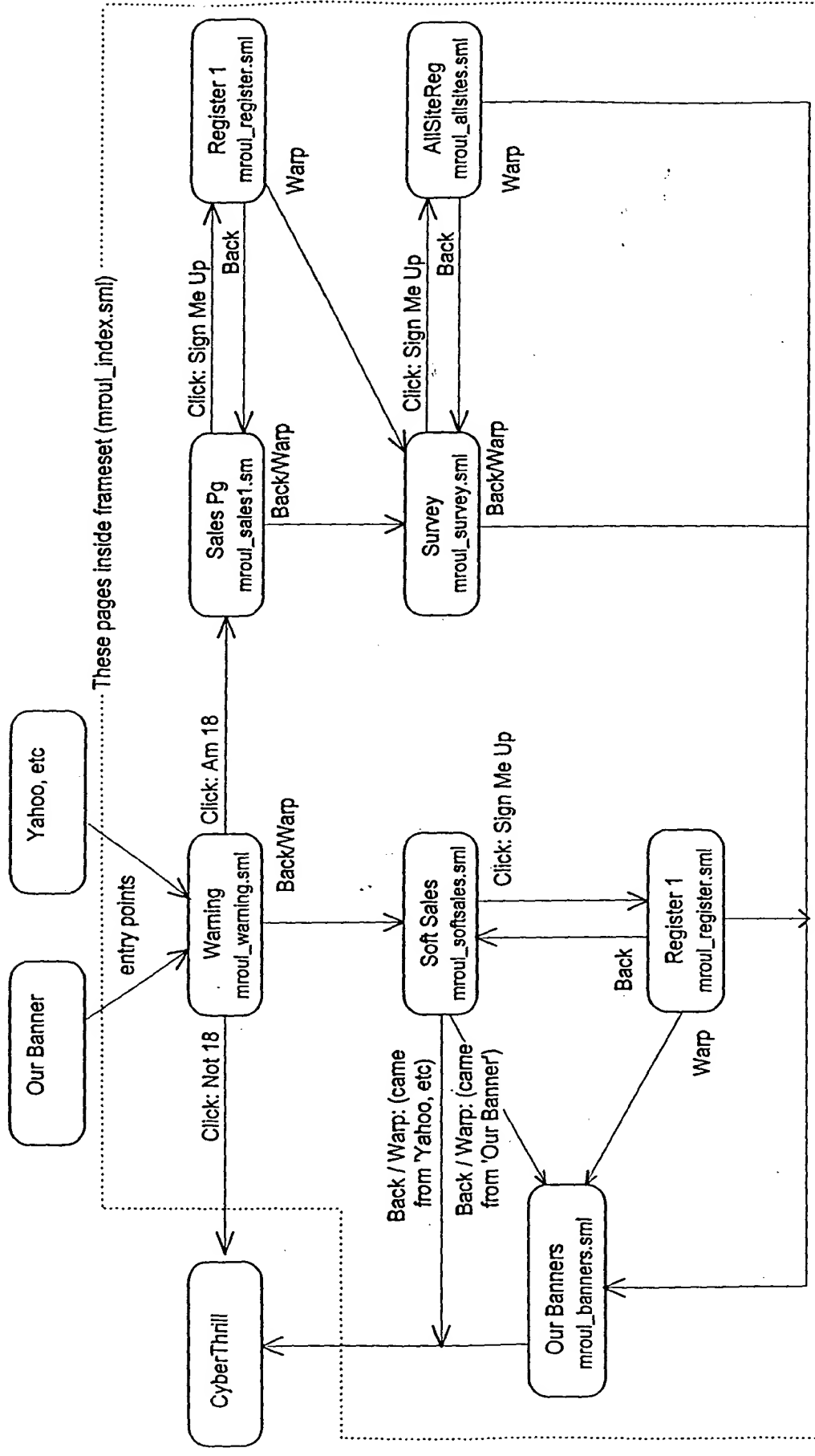
1. There will be a referrer in VSC that will indicate the user came from our own banners page (via back/warp of course)
2. The SML "MAC.visit" will not be used on the sales page.

The top of the "vsc_sales.sml" page should then have the following conditional:

```
<$if cond:.refer val:SID.referrer=.refer>
<$else>
<#MAC.visit#>
<!-- VH -->
<$endif>
```

On the "vsc_sales.sml" page, we don't want to cause a visit to be registered to a banner ID if the user is jumping to it via the banners page.

7. Mousetrap Flow Drawing -- Flow Chart for Sex Roulette



Sex Museum and Ass Awards are similar except there are 4 registration pages off the Sales Pages.